

Matlab component

Creating a component to handle electrical signals

OptiSystem Application Note



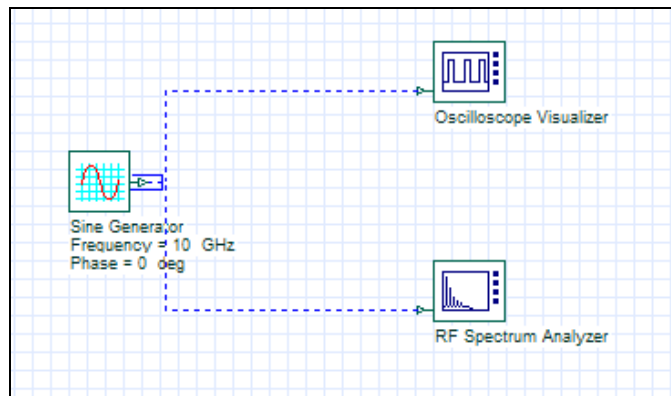
Matlab component – Creating a component to handle electrical signals

1. Electrical Attenuator Component

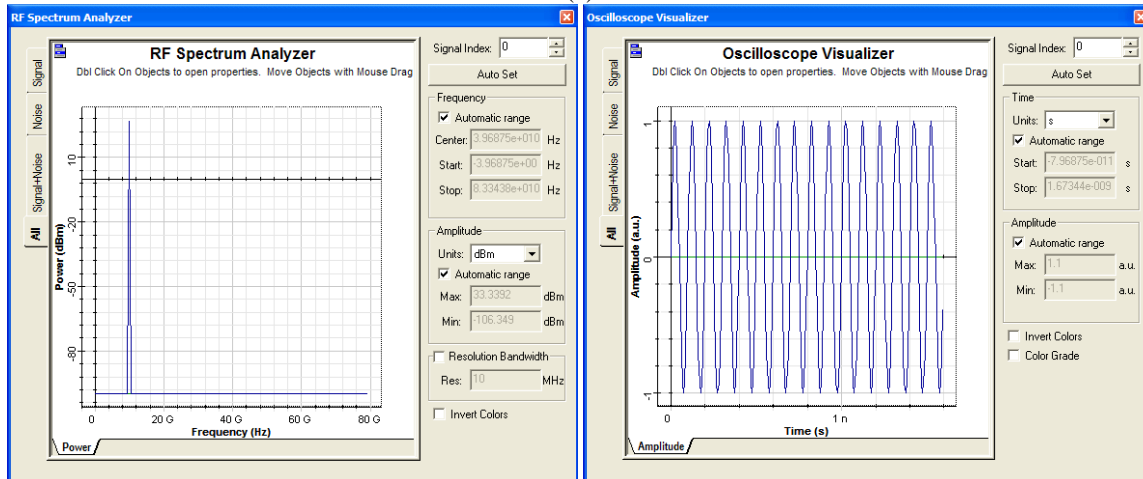
In order to create an electrical component in Matlab for co-simulation with OptiSystem, first we need to understand the electrical signal format that OptiSystem can generate and the structure of that signal launched into the Matlab workspace.

Following is an example to create an *Electrical Attenuator* using the Matlab component. In this example, first we introduce the signal format in OptiSystem, and then we show how to use Matlab to process that signal.

Electrical Sampled Signal – Figure 1 demonstrates the system layout for an electrical Sine Generator. Using an Oscilloscope Visualizer and a RF Spectrum Analyzer, the electrical sampled signal can be viewed in both frequency and time domain.



(a)

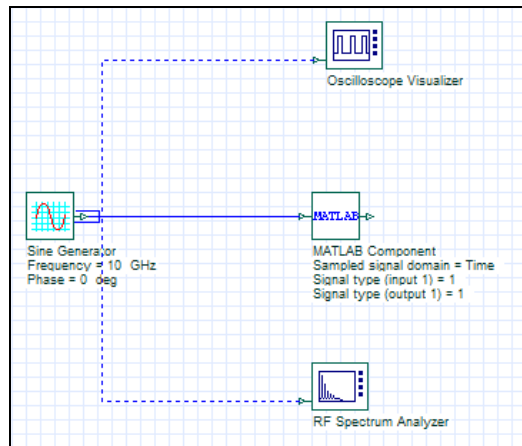


(b)

(c)

Figure 1: System layout and optical signal in frequency domain (spectrum) and time domain.

Using the MATLAB Library, we can add a Matlab component to the layout. By clicking on the component properties, on the Main tab, you can choose the Sampled Signal in the time or frequency domain. The number of inputs and outputs ports and also the format of these signals (Optical/Electrical) can be defined in the Inputs and Outputs tab. The User Parameter tab is used to define the input parameters of the component. The system layout and the Matlab component properties are shown in Figure 2.



MATLAB Component Properties

Label: MATLAB Component Cost\$: 0.00

OK Cancel Evaluate Script

Main Inputs Outputs User Parame... Simulation Random nu...

Disp	Name	Value	Units	Mode
<input type="checkbox"/>	Load Matlab			Normal
<input type="checkbox"/>	Run command	ElectricalAttenuator		Normal
<input type="checkbox"/>	Matlab search path	c:\Temp\		Normal
<input checked="" type="checkbox"/>	Sampled signal domain	Time		Normal
<input type="checkbox"/>	Spatial mode domain	Space		Normal

MATLAB Component Properties

Label: MATLAB Component Cost\$: 0.00

OK Cancel Evaluate Script

Main **Inputs** Outputs User Parame... Simulation Random nu...

Disp	Name	Value	Units	Mode
<input type="checkbox"/>	Number of input ports	1		Normal
<input checked="" type="checkbox"/>	Signal type (input 1)	1		Normal

MATLAB Component Properties

Label: MATLAB Component Cost\$: 0.00

OK Cancel Evaluate Script

Main Inputs **Outputs** User Parame... Simulation Random nu...

Disp	Name	Value	Units	Mode
<input checked="" type="checkbox"/>	Number of output ports	1		Normal
<input checked="" type="checkbox"/>	Signal type (output 1)	1		Normal

MATLAB Component Properties

Label: MATLAB Component Cost\$: 0.00

OK Cancel Evaluate Script

Main Inputs Outputs **User Parame...** Simulation Random nu...

Disp	Name	Value	Units	Mode
<input type="checkbox"/>	Number of parameters	1		Normal
<input type="checkbox"/>	Parameter0	0		Normal

Figure 2: System Layout and MATLAB Component Properties

Figure 3 shows the structure of the launched signal in the Matlab workspace. Each signal is defined by the sampled signal structure and the noise structure, however since there is no noise source in this example, the noise values are zero.

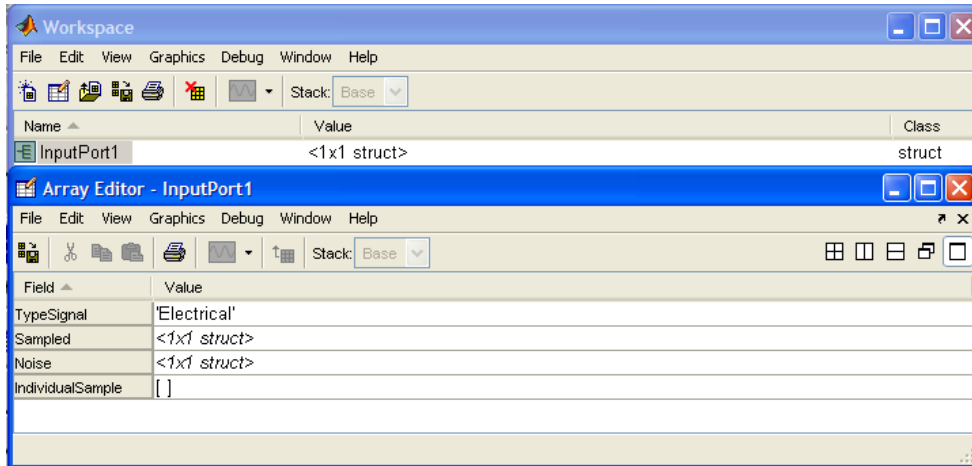


Figure 3: Signal structure launched into the Matlab workspace.

As it was mentioned before, the signal can be launched into the Matlab workspace in time or frequency domain (Matlab component parameter '*Sampled Signal Domain*'). Figure 4 demonstrates the Samples signal in Frequency and time domain respectively. The noise structure follows the same behavior.

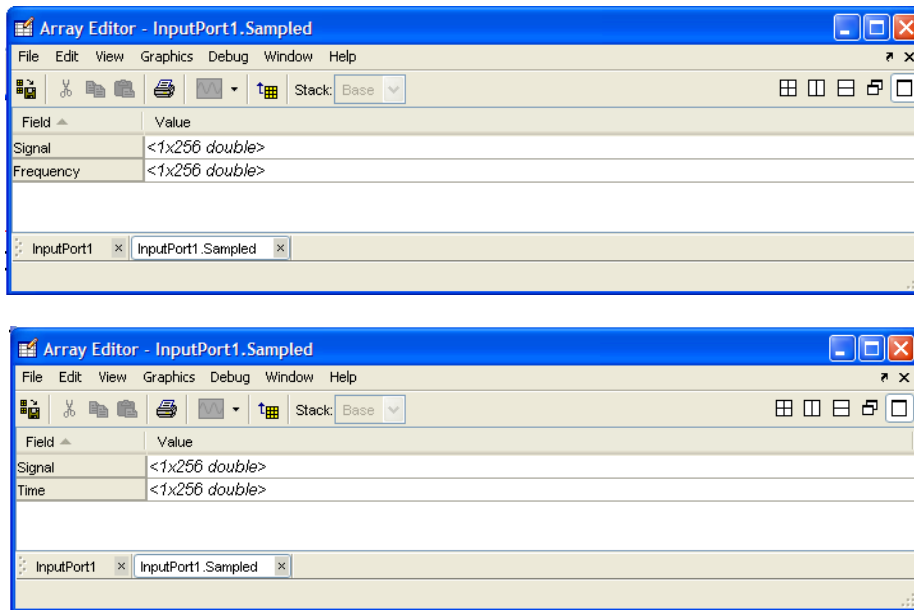


Figure 4: Sampled signal structure for the signal in frequency and time domain.

Unlike optical signals, the electrical sampled structure has only one dimension and all electrical signals are allocated in the same structure. In the following example, two electrical signals are generated and launched into the same input port. Figure 5 demonstrates the system layout.

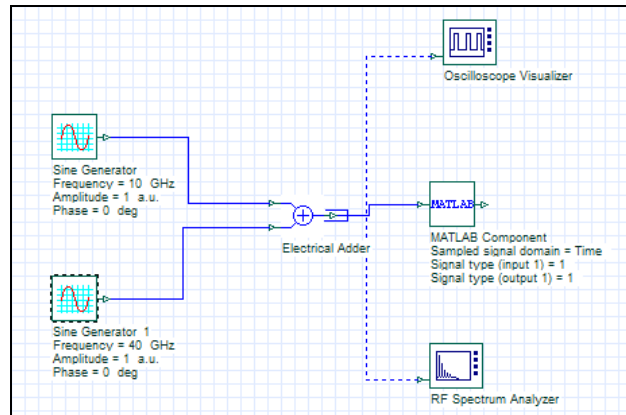
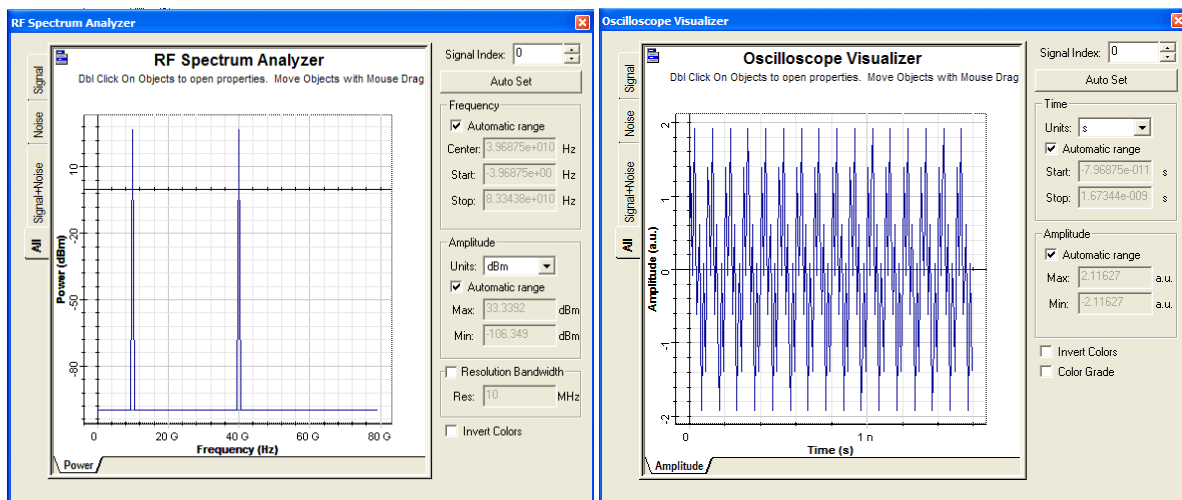


Figure 5: System layout.

Figure 6 shows the electrical signal launched into the Matlab workspace in time and frequency domain and Figure 7 is the structure of this signal in Matlab workspace.



(a)

(b)

Figure 6: Electrical signal in the (a) spectrum domain and (b) time domain.

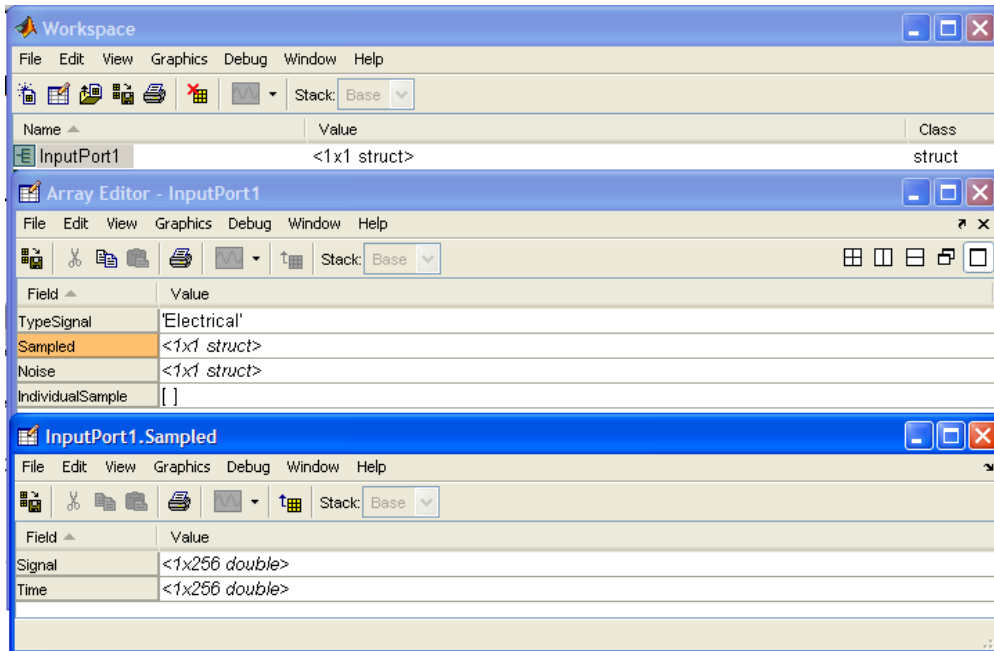


Figure 7: Structure launched into the workspace and electrical sampled signal structure.

Electrical Individual Sample Signal – Individual samples allow users to simulate time-driven systems in the electrical and optical domain. By using it, users can create designs that have closed loops and feedbacks. To convert an electrical sampled signal to Individual sample signals, the component “Convert To Electrical Individual Samples” (Signal Processing Library/Tools/Electrical) can be used. The system layout that launches electrical Individual samples into Matlab is presented in Figure 8.

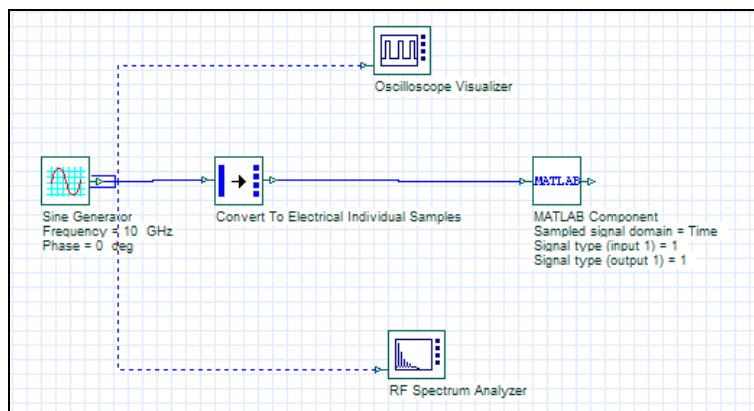
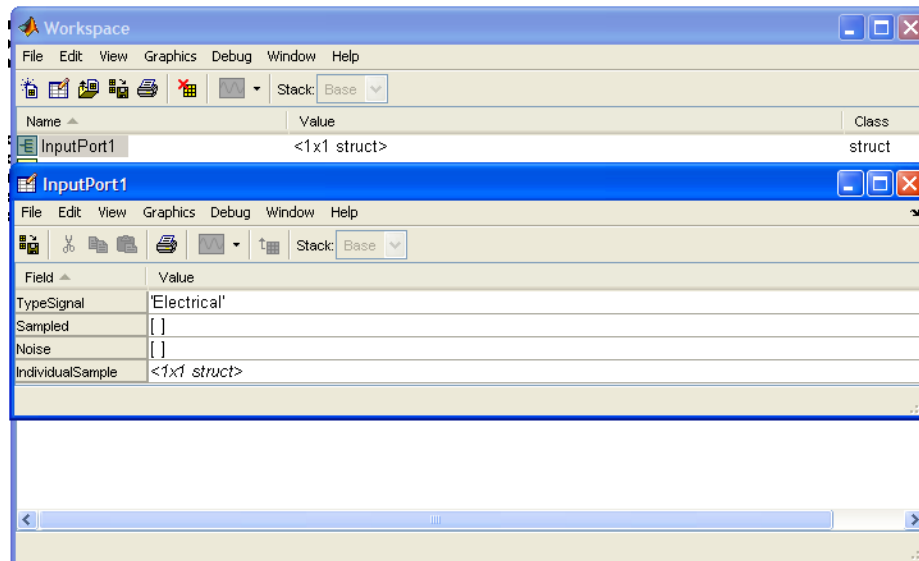
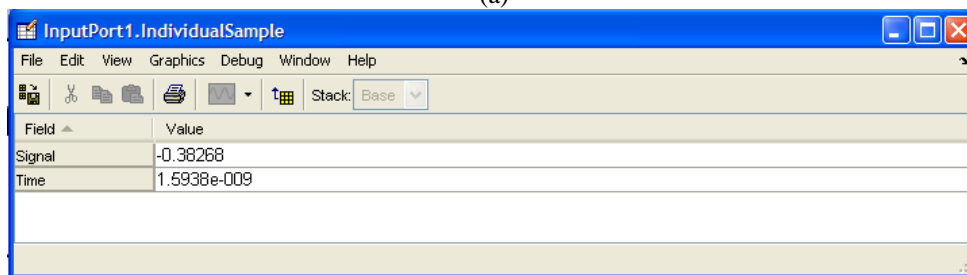


Figure 8: System layout with Individual Samples signal.

The structure launched in to the Matlab workspace is shown in Figure 9. OptiSystem put sample by sample in the Matlab workspace and the structure contains amplitude of the signal in a specified time. The sample showed in Figure 9.b is the last sample launched into Matlab workspace.



(a)



(b)

Figure 9: (a) Signal structure launched into the workspace, (b) Individual sample structure.

1.1 Matlab program:

Based on the signal structure, an electrical attenuator is implemented using the Matlab component. In this case the matlab component is set to have one input port and one output port for electrical signals. The Matlab program (m-file) that will simulate the attenuator is named *ElectricalAttenuator* and that is the command at 'Run command' parameter field. To be able to run this program, the path for this file must be in the Matlab search path, in this example the *ElectricalAttenuator.m* is located at 'c:\temp'.

In order to see the file structure and the m-file, go to the Matlab component properties, on the main tab, check 'Load Matlab' box and click OK. This will open the Matlab Command Window. In this window, first choose the code directory by 'cd ('c:\temp')' command. Using the command 'open *ElectricalAttenuator.m*', you can open the m-file in the Matlab Editor. The command 'workspace' opens the Matlab workspace.

The only parameter necessary for this component is the attenuation constant, which can be defined by the 'Parameter0' in the 'User Parameters tab'. More parameters can be added by using 'Add Parameter' button.

After defining the Matlab component properties, you can start writing the program (Matlab code) that simulates the electrical attenuator. Following is the Matlab code:

```

% This program simulates an electrical attenuator
% Input parameter - Attenuation constant [dB]

% create an output structure similar to the input
OutputPort1 = InputPort1;

% Attenuation constants
Attenuation = Parameter0;

if(InputPort1.TypeSignal == 'Electrical')

    cs = length(InputPort1.Sampled);

    if( cs > 0 )
        % Calculate the signal attenuated
        OutputPort1.Sampled.Signal = InputPort1.Sampled.Signal * exp(-0.2303 * Attenuation/2);
    end

    cp = length(InputPort1.Noise);

    if( cp > 0 )
        % Calculate the noise attenuated
        OutputPort1.Noise.Signal = InputPort1.Noise.Signal * exp(-0.2303 * Attenuation/2);
    end

    cn = length(InputPort1.IndividualSample);

    if( cn > 0 )
        % Calculate the signal attenuated
        OutputPort1.IndividualSample.Signal = InputPort1.IndividualSample.Signal * exp(-0.2303 *
Attenuation/2 );
    end
end
end

```

In this example, the code was written for the 'Sampled', 'Noise' and 'IndividualSample', however, according to the application, the user can define those ones required for the simulation, e.g. Sampled signal.

In order to see the file structure in the Matlab workspace (Figure 3), first you need to run the OptiSystem program, then use the command 'open InputPort1' in the Matlab Command Window, to see the file structure for each port. The command 'workspace' opens the Matlab workspace.

Please refer to the "Creating a component to handle optical signals" application note for the instructions of the Matlab debug mode.



Optiwave
7 Capella Court
Ottawa, Ontario, K2E 7X1, Canada

Tel.: 1.613.224.4700
Fax: 1.613.224.4706

E-mail: support@optiwave.com
URL: www.optiwave.com
Forum: www.optiwave.us